第 1 章　关于 LaTeX 你不得不知道的事

Du Ang
du2ang233@gmail.com

2017 年 7 月 17 日

目录

LaTeX 作者：Donald E. Knuth(高德纳)
LaTeX 读音："Lay-tech"或者"Lah-tech"
LaTeX 2ε读作："Lay-tech two e"
WYSIWYG: *What you see is what you get.* "所见即所得"。当前大多数文字处理软件的理念，如 MS Word、LibreOffice。但是 LaTeX 并没有遵循这个理念。

1 LaTeX 的优缺点

When people from the WYSIWYG world meet people who use LaTeX, they often discuss "the advantages of LaTeX over a normal word processor" or the opposite. The best thing to do when such a discussion starts is to keep a low profile(保持低调), since such discussions often get out of hand. But sometimes there is no escaping...

1.1 优点

- 可以获得专业的布局，让文档宛如印刷品一般精美
- 方便地排版数学公式
- 只需学几个简单的命令就能明确文档的逻辑结构，几乎不用对文档的布局修修补补
- 可以方便地生成脚注(footnotes)、引用(references)、目录(table of contents)、参考文献(bibliographies)等 [1]
- 拥有许多针对各种排版设计任务的免费附加宏包
- 促使用户写出结构良好的文档——而这也是 LaTeX 存在的初衷
- 跨平台，完全免费，容易获得

1.2 缺点

- 对于那些已经出卖了灵魂的人来说，LaTeX 可能会不好使
- 排查错误困难。LaTeX 作为一个依靠编写代码工作的排版工具，其使用的宏语言比 C++ 或 Python 等程序设计语言在错误排查方面困难得多。它虽然能够提示错误，但不提供调试的机制，有时错误提示还很难理解
- 在一个提前设计好的文档布局里改参数很容易，但设计一个新的布局比较困难，需要花费较多的时间
- 相比"所见即所得"的模式有一些不便，为了查看生成的文档，用户总要不停地编译
- 若是浅尝辄止，终不能领悟"逻辑标记"的真谛（ *Your hamster might, despite some encouraging first steps, never be able to fully grasp the concept of Logical Markup* ）

[1] Difference Between Bibliography and References: http://www.differencebetween.net/language/difference-between-bibliography-and-references/

2 LaTeX 输入文件

LaTeX 的输入文件是纯文本文件。

2.1 空格

- “Whitespace”字符，如空白、Tab 符等都会被 LaTeX 认为是空格
- 几个连续的空格会被 LaTeX 当作一个空格
- 每一行开头的空格会被忽略，末尾的断行符 `\\` 会被当作空格
- 一个空行是一个段落结束的标志
- 几个空行视为一个空行

2.2 特殊字符

保留字符： `#` `$` `^` `&` `_` `{` `}` `~` `\`

打印保留字符：`\#` `\$` `\^` `\&` `\_` `\{` `\}` `\~` `\textbackslash`

注意：保留字符有特殊含义，任何字体下直接输入都不会打印，但可以通过加反斜杠 `\` 的方式打印这些特殊字符。不能通过 `\\` 来输入 `\`，而应该使用 `\textbackslash`，`\\` 用于断行。

2.3 LaTeX 命令

LaTeX 命令是区分大小写的。LaTeX 命令以反斜线 `\` 开头，为以下两种形式之一：

- 反斜线和后面的一串字母，如 `\LaTeX`。它们以任意非字母符号（空格、数字、标点等）作为分隔符。如`\textsl{}`、`\newline`命令。
- 反斜线和后面的一个非字母符号，如 `\$`，它们之间无需分隔符。

LaTeX 会忽略掉命令后的空白符。如果想在命令后添加空格，可以在命令后添加一个空的 `{}`，如 `\TeX{}`；或者是在命令后添加一个空格命令或特殊字符（如 `$\sim$` ）。

大多数的 LaTeX 命令是带一个或多个参数，每个参数用花括号 `{` 和 `}` 包裹。有些命令带一个或多个可选参数，以方括号 `[` 和 `]` 包裹，用法为 `\command[optional parameter]{parameter}`。

还有些命令在命令名称后可以带一个星号 `*`，带星号和不带星号的命令效果有一定差异。

LaTeX 还引入了环境的用法，用以令一些效果在局部生效，或是生成特定的文档元素。LaTeX 环境的用法为一对命令 `\begin` 和 `\end`。

2.4 LaTeX 注释

利用 `%` 进行单行注释，引入 `verbatim` 宏包（`\usepackage{verbatim}`）后，可以在 `\begin{comment}` 和 `\end{comment}` 之间加入多行注释。

3 LaTeX 输入文件结构

```
\documentclass{...}
\usepackage{...}

\begin{document}
    % text with some LaTeX commands
\end{document}
```

4 LaTeX 布局

典型命令：`\documentclass[options]{class}`
class 指定文档类型，包含以下几种类型：

- article 文章格式的文档类，广泛用于科技论文、报告、说明文档等
- report 长篇报告格式的文档类，具有章节结构，用于综述、长篇论文、简单的书籍等
- book 书籍文档类，包含章节结构和前言、正文、后记等结构
- proc 基于 article 文档类的一个简单的学术文档模板
- slides 幻灯格式的文档类，使用无衬线字体
- minimal 一个极其精简的文档类，只设定了纸张大小和基本字号，用作代码测试的最小工作示例

options 为文档类指定选项，以全局地影响文档布局的参数，如字号、纸张大小、单双面等等。比如调用 article 文档类排版文章，指定纸张为 A4 大小，基本字号为 11pt。
双面排版：`\documentclass[11pt,twoside,a4paper]{article}`。

LaTeX 的三个标准文档类可指定的选项：

- 10pt, 11pt, 12pt 指定文档的基本字号。缺省为 10pt
- a4paper, letterpaper, ... 指定纸张大小，默认为美式纸张 letterpaper。可指定选项还包括 a5paper、b5paper、executivepaper 和 legalpaper
- fleqn 令行间公式左对齐 (缺省为居中)
- leqno 将公式编号放在左边 (缺省为右边)
- titlepage, notitlepage 指定标题命令。 是否生成单独的标题页。article 缺省为 notitlepage，report 和 book 缺省为 titlepage
- onecolumn, twocolumn 指定单栏/双栏排版
- twoside, oneside 指定单面/双面排版。双面排版时，奇偶页的页眉页脚、页边距不同。article 和 report 缺省为单面排版，book 缺省为双面。
- landscape 指定横向排版。缺省为纵向。

- openright, openany 指定新的一章 \chapter 是在奇数页 (右侧) 开头，还是直接紧跟着上一页开头。report 缺省为 openany，book 缺省为 openright。(对 article 无效)

LaTeX 支持使用 \pagestyle{style}来定义页眉页脚的风格，有三种预定义的选项可以选择：

- plain 默认为此选项，在每一页的页脚中间打印当前页码
- headings 在每一页的页眉处打印当前章节（chapter）名称和当前页码，页脚为空
- empty 页眉页脚都设为空
- myheadings 没有页脚，页眉是页码和用户自定义的内容

可以使用\thispagestyle{style}来改变特定页的页眉页脚风格。

5 LaTeX 用到的文件一览

除了源码.tex 文件，还可能遇到以下格式的文件：

- .sty 宏包文件。宏包的名称就是去掉扩展名的文件名
- .cls 文档类文件。同样地，文档类名称就是文件名
- .bib BibTeX 参考文献数据库文件
- .bst BibTeX 用到的参考文献格式模板

LaTeX 在编译过程中生成相当多的辅助文件和日志。一些功能如交叉引用、参考文献、目录、索引等，需要先编译生成辅助文件，然后再次编译时读入辅助文件得到正确的结果，所以复杂的 LaTeX 源代码可能要编译多次。在使用 TeX 的过程中可能遇到以下扩展名的文件：

- .log 排版引擎生成的日志文件，供排查错误使用
- .aux LaTeX 生成的主辅助文件，记录交叉引用、目录、参考文献的引用等
- .toc LaTeX 生成的目录记录文件
- .lof LaTeX 生成的图片目录记录文件
- .lot LaTeX 生成的表格目录记录文件
- .bbl BibTeX 生成的参考文献记录文件
- .blg BibTeX 生成的日志文件
- .idx LaTeX 生成的供 makeindex 处理的索引记录文件
- .ind makeindex 处理.idx 生成的格式化索引记录文件
- .ilg makeindex 生成的日志文件
- .out hyperref 宏包生成的 PDF 书签记录文件

6 文件的组织方式

当编写较大规模的 LaTeX 源代码，如书籍、毕业论文等，你有理由将源代码分成若干个文件而不是写到一堆，比如很自然地每章写一个文件。

可以使用 `\include` 命令在源代码中插入文件，如 `\include{⟨filename⟩}`。⟨filename⟩ 为文件名（如果是 `.tex` 文件，可以不加扩展名），如果和要编译的主文件不在一个目录中，则要加上相对或绝对路径。值得注意的是 `\include` 在读入 ⟨filename⟩ 之前会另起一页。有的时候我们并不需要这样，而是用 `\input` 命令，它纯粹是把文件里的内容插入: `\input{⟨filename⟩}`。

另外 LaTeX 提供了一个 `\includeonly` 命令来组织文件，用于导言区（preamble），指定只载入某些文件: `\includeonly{⟨filename1⟩, ⟨filename2⟩, ...}`。导言区使用了 `\includeonly` 后，正文中不在其列表范围的 `\include` 命令不会起效。

注意：参考 *TEX and controlled access to information*[2]，LaTeX 中没有 `\inputonly` 命令，但是可以自己定义一个：

```
\newif\iffinancial
...
\iffinancial\input{cost_table.tex}\fi
```

7 编译小技巧

加载宏包 `syntonly`，在导言区使用 `\syntaxonly` 命令, 可令 LaTeX 编译后不生成 DVI 或者 PDF 文档, 只排查错误, 编译速度会快不少：

```
\usepackage{syntonly}
\syntaxonly
```

如果想生成文档, 则将 `\syntaxonly` 命令那一行用% 注释掉即可。如果在编译 `.tex` 文件时遇到了错误，使用 `Ctrl+D` 回到命令行。

[2] https://www.tug.org/TUGboat/tb36-2/tb113veytsman-access.pdf